Supplementary Material for CountTRuCoLa: Rule Confidence Learning for Temporal Knowledge Graph Forecasting

Julia Gastinger, Christian Meilicke, Heiner Stuckenschmidt
University of Mannheim
{first.last}@uni-mannheim.de

1 Special remark on c-rules

In this section we are providing a more detailed consideration of c-rules with the help of an example. Consider a dataset in which a regularity is observed: if a person eats pizza at timestamp t, they will drink espresso at a subsequent timestamp. This pattern can be represented by the following c-rule, where e denotes eat and d denotes drink:

$$d(x, espresso, t^*) \leftarrow e(x, pizza, t) \wedge\ t^* > t \qquad (1)$$

This c-rule can be directly applied, as described in Section 4.2, to answer a query such as $drinks(x, ?, t^*)$, to determine what person x, who just ate pizza, will drink.

The mentioned regularity can also be used to answer a different type of query: Who will drink an espresso at timestamp t^*, expressed as $drinks(?, espresso, t^*)$, or equivalently, using inverse relations, as $drinks^{-1}(espresso, x, t^*)$.

Intuitively this could be expressed by such a rule:

$$d^{-1}(espresso, x, t^*) \leftarrow e^{-1}(pizza, x, t) \wedge\ t^* > t \qquad (2)$$

However, our language bias does not support rules where the constant appears in the subject position, meaning Rule (2) is not covered by our language bias.

Luckily, there is an inverse relation for each relation. This means that we can express the regularity as

$$d(x, espresso, t^*) \leftarrow e(x, pizza, t) \wedge\ t^* > t \qquad (3)$$

Rule (3) is covered by our language bias.

The fact that Rule (3) looks the same as Rule (1) is because so far we have ignored an important detail: We have omitted the additional body atom, which ensures that the query is asked only if there is at least one correct answer. It makes a difference whether such a rule should answer queries such as $drinks(x, ?, t^*)$, that we call forward queries, or queries such as $drinks(?, espresso, t^*)$, which we call backward queries. For the forward query the additional body atom is $drinks(x, z, t^*)$, for the backward query $drinks(z, espresso, t^*)$,

This body atom does not only make a difference in the formal writing of the rule, but also when collecting the examples E_r and learning the parameters for the rule. For both rules, we need to find cases where people drink espresso after eating pizza as positive examples. However the negative examples differ: For the first query, $drinks(x, ?, t^*)$, people who drink tea after eating pizza would be negative examples. For the second case, $drinks(?, espresso, t^*)$, the negative example would be if one person eats pizza and in the next timestamp, another person drinks espresso.

Thus, to support any possible application of a c-rule, we have to compute two variants of the same c-rule that differ only with respect to the additional atom:

$$h(x, d, t^*) \leftarrow b(x, d', t) \wedge \exists z\ h(x, z, t^*) \wedge\ t^* > t \qquad (4)$$

$$h(x, d, t^*) \leftarrow b(x, d', t) \wedge \exists z\ h(z, d, t^*) \wedge\ t^* > t \qquad (5)$$

where Rule (4) belongs to the query such as $drinks(x, ?, t^*)$, and Rule (5) belongs to the query such as $drinks(?, espresso, t^*)$. We call Rule (4) the forward version of a c-rule (c-rule F), and Rule (5) the backward version of a c-rule (c-rule B).

Depending on the query that we have to answer, we use the appropriate variant of the relevant c-rules. In Line 6 in Algorithm 1 (main paper) we have to use the additional atom of the forward to compute the example set for the first variant, or the additional atom of the backward variant to compute the example set for the second variant.

2 Additional Information on Learning the Confidence Functions

As described in Section 4, *Learning the Confidence Functions*, of the main paper, we transform the observed confidence value by introducing a scaling factor that depends on the minimal recency distance. This transformation is controlled by a hyperparameter $\mathcal{P}$, which is added to the denominator when computing the confidence.

Below, we provide a detailed description of this transformation: For each group of examples in E_r sharing the same minimal recency distance $\min(\Delta_i)$, we compute a scaling factor $k(\min(\Delta_i))$ and use it to adjust the observed confidence value y_i. The transformation is defined as:

$$\tilde{y} = s(\Delta, y) = k(\min(\Delta)) \cdot y, \qquad (6)$$

The following pseudo-code (Algorithm 1) outlines the computation of k:

Algorithm 1: Computing the scaling factor k

Input: Example Set E for rule r, Parameter $\mathcal{P}$
Output: Mapping k from minimal distance d to scaling factor
Initialize empty set $X = \emptyset$ and empty maps p, a, k
for each (Δ_i, y_i) in E **do**
    $d = \min(\Delta_i)$ // minimal distance
    if $d \notin X$ **then**
        Add d to X, $p(d) = 0$, $a(d) = 0$
    $p(d) = p(d) + y_i$ // count positive examples for d
    $a(d) = a(d) + 1$ // count all examples for d
for each d in X **do**
    $k(d) = \frac{a(d)}{a(d)+\mathcal{P}}$ // compute scaling factor for d
return k

This approach ensures that the observed confidence values are adjusted according to the number of examples for each minimal recency distance, controlled by the hyperparameter $\mathcal{P}$.

3 Additional Information on Datasets

We provide additional details on the datasets used in our experiments. For each dataset, we use the version provided by (Li et al. 2021) and (Gastinger et al. 2023), or, if lowercase letters only, the datasets introduced by (Gastinger et al. 2024). An overview on the statistics of the datasets is in Table 1.

ICEWS Datasets: `ICEWS14` (García-Durán, Dumančić, and Niepert 2018), `ICEWS18` (Jin et al. 2019), and `icews` (Gastinger et al. 2024) are derived from the Integrated Crisis Early Warning System (ICEWS) (Boschee et al. 2015; Shilliday, Lautenschlager et al. 2012). These datasets span different periods (2014, 2018, and 1995–2022) and contain event data on global political activities such as conflicts, protests, and diplomatic interactions. Events are categorized according to the CAMEO taxonomy (Gerner et al. 2002).

GDELT: The Global Database of Events, Language, and Tone (`GDELT`) (Leetaru and Schrodt 2013) contains large-scale event data extracted from global news sources. It encompasses a wide range of political, societal, and cultural events across various countries and timeframes.

polecat: Based on the POLECAT (POLitical Event Classification, Attributes, and Types) dataset (Scarborough et al. 2023), this dataset records cooperative and hostile interactions between socio-political actors. POLECAT uses the PLOVER ontology (Halterman et al. 2023) and automated NLP pipelines to classify and extract time-stamped, geolocated events from multilingual news sources. The dataset used in this work covers the period from January 2018 to December 2022.

YAGO and WIKI: `YAGO` (Mahdisoltani, Biega, and Suchanek 2015) and `WIKI` (Leblay and Chekol 2018) provide structured knowledge graph data with temporal relations. `WIKI` is extracted from Wikidata (Vrandečić and Krötzsch 2014) and both datasets have been further processed by (Jin et al. 2019) to represent temporal facts as quadruples. Events before 1786 (`WIKI`) and 1830 (`YAGO`) are excluded.

smallpedia and wikidata: These datasets are constructed from Wikidata (Vrandečić and Krötzsch 2014) and processed by (Gastinger et al. 2024). `smallpedia` includes entities with IDs below 1 million, while `wikidata` extends the scope to entities with IDs up to 32 million. Both datasets contain event-based (point-in-time) and fact-based (duration) temporal relations between entities.

4 Details on Experimental Setup

4.1 Reasons for Excluding Prior Methods from Comparison

As noted in Section 5.1 of the main paper, we exclude 15 prior methods from direct comparison due to various limitations. In this section, we provide detailed justifications for each exclusion.

INFER reports result on the "best" ranking protocol in the presence of ties. As a result, the evaluation always assigns the best possible rank to the ground-truth entity in the case of ties, rather than using an average or random tie-breaking strategy. This protocol is known to inflate evaluation metrics unfairly, since it does not reflect the true ranking uncertainty in the presence of score ties (Sun et al. 2020). L2TKG, TPAR, Logenet, TempValid, ALREIR and TECHS do not provide code to reproduce results. CENET, RETIA, and CluSTER do not report results in time-aware filter setting. CyGNet and RE-Net run only in multi-step setting, not in single-step setting. TR-Rules uses different dataset versions for ICEWS14 and does not report results on WIKI, YAGO, GDELT, or the TGB 2.0 datasets. GenTKG evaluates on different versions of GDELT and YAGO, does not report results on WIKI or the TGB 2.0 datasets, and does not provide MRR scores for any dataset. Finally, we do not compare to zrLLM because it uses a different evaluation setup, focusing on zero-shot relations with different datasets and on predicting previously unseen relations.

4.2 Negative Samples

Following the TGB 2.0 evaluation framework, we use the provided negative samples for the large dataset `tkgl-wikidata`. Specifically, TGB 2.0 includes 1,000 negative samples per query, sampled based on the query relation type. As a result, evaluation on `tkgl-wikidata` is performed by ranking the correct entity against these 1,000 candidates rather than against all entities. For all other datasets, we compute scores by ranking against the full set of entities.

5 Hyperparameters

CountTRuCoLa comes with eight hyperparameters:

- $\mathcal{P}$: A constant added to the denominator when computing confidences for xy- and c-rules.
- $\mathcal{P}_{f\text{-rules}}$: The corresponding constant used for f-rules.
- $\mathcal{C}$: A boolean flag indicating whether c-rules are used for a given dataset.

Table 1: Dataset statistics. # Quads refers to the number of quadruples without inverse quadruples.

Dataset	smallpedia	polecat	icews	wikidata	ICEWS14	ICEWS18	GDELT	YAGO	WIKI
# Quads Train	387,757	1,246,556	10,861,600	6,982,503	74,845	373,018	1,734,399	161,540	539,286
# Quads Valid	81,033	266,736	2,326,157	1,434,950	8,514	45,995	238,765	19,523	67,538
# Quads Test	81,586	266,318	2,325,689	1,438,750	7,371	49,545	305,241	20,026	63,110
# Nodes	47,433	150,931	87,856	1,226,440	7,128	23,033	7,691	10,623	12,554
# Relations	283	16	391	596	230	256	240	10	24
# Timestamps	125	1,826	10,224	2,025	365	303	2,975	188	231
Granularity	year	day	day	year	day	day	15 min	year	year

- $\mathcal{W}$: The window size defining the length of the time window for the examples E, i.e., the maximum value in Δ.
- $\mathcal{M}$: A threshold defining the minimum number of data points required before learning the parameters of the frequency-based scoring function g_r. If a rule has fewer than $\mathcal{M}$ examples, all parameters of g_r are set to zero to reduce noise.
- $\mathcal{Z}$: A scaling factor $\mathcal{Z} \in [0, 1]$ applied to the score predicted by the z-rules.
- $\mathcal{H}$ and $\mathcal{D}$: Two hyperparameters for rule aggregation. $\mathcal{H}$ specifies how many of the top-confidence rules to aggregate, and $\mathcal{D} \in [0, 1]$ is a decay factor used in our modified noisy-or model, where the i-th confidence score s_i is weighted by $s_i \cdot \mathcal{D}^i$.

We perform grid search based on validation MRR to select the values of these hyperparameters for each dataset. For smaller datasets, we explore a broader range of values, while for larger datasets, we reduce the search space to limit runtime and energy consumption. Table 2 lists the tested ranges for each dataset. Note that the range for $\mathcal{W}$ varies depending on the total number of timesteps available in each dataset. Table 3 reports the hyperparameter values selected for each dataset.

6 Additional Results

In the following, we provide additional results.

6.1 Variance Across Repetitions

Table 4 reports test results across five repetitions for two selected datasets. We observe that the variance is less than 0.002 for all test metrics and datasets, which is small considering that the MRR and Hits@10 range from 0% to 100%. Compared to embedding-based methods, which may introduce randomness through factors such as random initialization or noise injection, our approach has fewer potential sources of variability: (i) sampling candidates for the examples of c-rules, (ii) computing the parameters that optimize the temporal confidence functions, and (iii) assigning random order to tied ranks during evaluation.

6.2 Significance Tests

We conduct a Chi-squared test to assess whether the observed differences in Hits@10 scores between methods are statistically significant. Since Hits@10 is a binary metric (a prediction is either in the top 10 or not), we treat the predictions as categorical outcomes: *hit* and *miss*. For a dataset, we have $2 \times N$ samples (where N is the number of test quadruples), accounting for inverse relations.

For example, if a model achieves a Hits@10 score of 0.6 on a test set with 100 quadruples, this corresponds to $0.6 \times 200 = 120$ *hit* predictions and 80 *miss* ones.

We compare CountTRuCoLa with the best-performing state of the art method regarding Hits@10 under the following null hypothesis:

> $\mathbf{H_0}$: Prediction correctness (i.e., *hit* vs. *miss*) is independent of the method used.

For examples, for the GDELT dataset, the Chi-squared test indicates a significant association between prediction correctness and method ($\chi^2 = 31.8$, $df = 1$, $p < 0.001$), suggesting that the improvement in Hits@10 by CountTRuCoLa over the baseline is statistically significant. Based on $p < 0.001$, in total, for the 5 datasets, where CountTRuCoLa has higher Hits@10 scores than the others, for 3 of them the Chi-squared tests suggests a statistically significant improvement. For the 4 datasets, where CountTRuCoLa has lower Hits@10 scores than the other, methods, for 3 of them the Chi-squared test suggests a statistically significant difference.

6.3 Runtime

Table 6 reports runtimes in seconds for CountTRuCoLa on all datasets. We report total time (incl. loading the dataset, and necessary steps), and times for the individual steps reported in the main paper, i.e., creating the Examples E for each rule (Section 4.2), learning the parameters for each rule (Section 4.2), rule application and aggregation (Section 4.3), and evaluation (Section 5). The runtimes were measured on a AMD EPYC 9474F 48-Core Processor. We did not use GPUs. Parallelization during rule application was handled via Ray, with a maximum of 20 concurrent threads.

For comparison, Table 7 shows the runtimes reported in (Gastinger et al. 2024) on the datasets presented in their study. These results were obtained on different hardware[1] and are therefore not directly comparable to our runtimes. However, they serve as a useful reference point. The table demonstrates that CountTRuCoLa's total runtimes for smallpedia and polecat are significantly lower than

[1]Experiments were conducted on Nvidia A100, V100, V100SXM2, and RTX8000 GPUs with 4 CPU nodes (from AMD Rome, Milan, or Intel Skylake) per experiment, using up to 1056 GB of RAM.

Table 2: Hyperparameter ranges. We allow fewer values for larger datasets to reduce computation costs.

	small WIKI, ICEWS14, YAGO, smallpedia	medium ICEWS18	large polecat, wikidata, GDELT	very large icews
$\mathcal{P}$ (RULE_UNSEEN_NEGATIVES)	$\{0, 1, 2, 3, 5, 10, 20, 30, 100\}$	$\{1, 5, 10, 30, 100\}$	$\{1, 10, 30, 100\}$	$\{1, 10, 30, 100\}$
$\mathcal{P}_{f\text{-rules}}$ (F_UNSEEN_NEGATIVES)	$\{0, 1, 5, 10, 20, 30\}$	$\{0, 10, 30\}$	$\{10\}$	$\{10\}$
$\mathcal{C}$ (RULE_TYPE_C)	$\{\text{True}, \text{False}\}$	$\{\text{True}, \text{False}\}$	$\{\text{True}, \text{False}\}$	$\{\text{False}\}$
$\mathcal{W}$ (LEARN_WINDOW_SIZE)	$\{50, 100, 150\}$ ICEWS14, $\{2, 3, 5, 10, 30, 50, 100\}$ WIKI, $\{2, 3, 5, 10, 30, 50, 100\}$ smallpedia, $\{2, 3, 5, 10, 30, 30, 50\}$ YAGO	$\{50, 100, 150\}$	$\{50, 100, 150\}$ polecat, $\{2, 3, 5, 10, 30, 50\}$ wikidata, $\{10, 30, 50, 100, 150, 200\}$ GDELT	$\{50, 100\}$
$\mathcal{M}$ (DATAPOINT_THRESHOLD_MULTI)	$\{0, 10, 50\}$	$\{0, 50\}$	$\{0, 50\}$	$\{50\}$
$\mathcal{Z}$ (Z_RULES_FACTOR)	$\{0, 0.1, 0.2, 0.3, 0.4, 0.5, 1.0\}$	$\{0, 0.1, 0.5\}$	$\{0, 0.1, 0.5\}$	$\{0, 0.1, 0.5\}$
$\mathcal{H}$ (NUM_TOP_RULES)	$\{5, 10, 50\}$	$\{10, 50\}$	$\{10\}$	$\{10\}$
$\mathcal{D}$ (AGGREGATION_DECAY)	$\{1, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4\}$	$\{1, 0.8, 0.6\}$	$\{0.8\}$	$\{0.8\}$

Table 3: Hyperparameter values for each dataset, selected based on the validation MRR.

	GDELT	YAGO	WIKI	ICEWS14	ICEWS18	smallp.	polecat	icews	wikidata
$\mathcal{P}$ (RULE_UNSEEN_NEGATIVES)	30	30	1	30	100	3	30	100	30
$\mathcal{P}_{f\text{-rules}}$ (F_UNSEEN_NEGATIVES)	10	30	10	10	10	30	10	10	10
$\mathcal{C}$ (RULE_TYPE_C)	False	True	True	True	True	True	False	False	True
$\mathcal{W}$ (LEARN_WINDOW_SIZE)	200	30	3	50	50	3	150	50	10
$\mathcal{M}$ (DATAPOINT_THRESHOLD_MULTI)	0	50	50	0	50	10	0	50	0
$\mathcal{Z}$ (Z_RULES_FACTOR)	0.1	0.1	0.1	0.1	0.1	0	0.1	0.1	0.1
$\mathcal{H}$ (NUM_TOP_RULES)	10	5	10	10	10	5	10	10	10
$\mathcal{D}$ (AGGREGATION_DECAY)	0.8	0.7	0.8	0.8	0.8	0.4	0.8	0.8	0.8

Table 4: Test results across five repetitions for datasets WIKI and ICEWS14, as well as mean and variance.

	WIKI		ICEWS14	
	MRR	H10	MRR	H10
	82.6809	86.5592	44.9916	62.0201
	82.6615	86.5568	44.9828	62.0404
	82.5704	86.5647	45.0014	62.1083
	82.6181	86.5624	44.9864	62.0336
	82.5992	86.5584	44.9685	62.0201
Mean	82.6260	86.5603	44.9862	62.0445
Variance	0.002038	0.000010	0.000146	0.001348

Table 5: Chi-squared test for all datasets. We test for $(df = 1, p < 0.001)$. and compare CountTRuCoLa with the best performing state of the art method regarding Hits@10.

Dataset	CountTRuCoLa H10	Best Baseline H10	Best Baseline	χ^2	Significant?
Group 1: Rucola best method					
GDELT	40.3	39.8	Rec. B.	31.78	yes
YAGO	93.2	93.0	Rec. B.	1.25	no
smallpedia	71.7	71.6	Rec. B.	0.40	no
polecat	40.8	37.8	Tlogic	1004.76	yes
wikidata	62.8	59.6	Edgebank	6204.43	yes
Group 2: Rucola not best method					
WIKI	86.6	87.1	Rec. B. & TiRGN	13.81	yes
ICEWS14	62.0	63.8	TiRGN	10.23	no
ICEWS18	51.0	54.2	TiRGN	203.49	yes
icews	32.1	33.4	CEN	1784.58	yes

those of related work, despite not using a GPU. For `icews`, it achieves comparable runtimes to other methods, and for `wikidata`, it is the only method aside from the simple Edgebank heuristic that produces results.

6.4 Memory Limits

The upper memory limits set in SLURM for each dataset (using 20 parallel processes) are as follows:

- Small datasets (`WIKI`, `ICEWS14`, `YAGO`, `tkgl-smallpedia`): 20 GB
- `polecat`: 160 GB
- `ICEWS18`: 300 GB
- `gdelt`: 350 GB
- `wikidata`: 200 GB
- `tkgl-icews`: 500 GB

Please note that these are upper limits, not actual memory usage. Additionally, reducing the number of parallel processes during the application phase will decrease memory consumption at the cost of increased runtime.

References

Boschee, E.; Lautenschlager, J.; O'Brien, S.; Shellman, S.; Starz, J.; and Ward, M. 2015. ICEWS Coded Event Data.

García-Durán, A.; Dumančić, S.; and Niepert, M. 2018. Learning Sequence Encoders for Temporal Knowledge Graph Completion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 4816–4821. Brussels, Belgium: Association for Computational Linguistics.

Gastinger, J.; Huang, S.; Galkin, M.; Loghmani, E.; Parviz, A.; Poursafaei, F.; Danovitch, J.; Rossi, E.; Koutis, I.; Stuckenschmidt, H.; Rabbany, R.; and Rabusseau, G. 2024. TGB 2.0: A Benchmark for Learning on Temporal Knowledge Graphs and Heterogeneous Graphs. In *38th Conference on Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*.

Gastinger, J.; Sztyler, T.; Sharma, L.; Schuelke, A.; and Stuckenschmidt, H. 2023. Comparing Apples and Oranges? On the Evaluation of Methods for Temporal Knowledge Graph Forecasting. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD)*, 533–549.

Gerner, D. J.; Schrodt, P. A.; Yilmaz, O.; and Abu-Jabr, R. 2002. Conflict and mediation event observations (cameo): A new event data framework for the analysis of foreign policy interactions. *International Studies Association, New Orleans*.

Halterman, A.; Bagozzi, B. E.; Beger, A.; Schrodt, P.; and Scraborough, G. 2023. PLOVER and POLECAT: A new political event ontology and dataset. In *International Studies Association Conference Paper*.

Jin, W.; Qu, M.; Jin, X.; and Ren, X. 2019. Recurrent event network: Autoregressive structure inference over temporal knowledge graphs. *arXiv preprint arXiv:1904.05530*. Preprint version.

Leblay, J.; and Chekol, M. W. 2018. Deriving Validity Time in Knowledge Graph. In Champin, P.; Gandon, F.; Lalmas, M.; and Ipeirotis, P. G., eds., *Companion of the The Web Conference 2018 on The Web Conference 2018, WWW 2018, Lyon, France, April 23-27, 2018*, 1771–1776. ACM.

Leetaru, K.; and Schrodt, P. A. 2013. Gdelt: Global data on events, location, and tone, 1979–2012. In *ISA annual convention*, 1–49. Citeseer.

Li, Z.; Jin, X.; Li, W.; Guan, S.; Guo, J.; Shen, H.; Wang, Y.; and Cheng, X. 2021. Temporal Knowledge Graph Reasoning Based on Evolutional Representation Learning. In *The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*.

Mahdisoltani, F.; Biega, J. A.; and Suchanek, F. M. 2015. YAGO3: A Knowledge Base from Multilingual Wikipedias. In *CIDR*.

Scarborough, G. I.; Bagozzi, B. E.; Beger, A.; Berrie, J.; Halterman, A.; Schrodt, P. A.; and Spivey, J. 2023. POLECAT Weekly Data.

Table 6: CountTRuCoLa Runtimes for all datasets in seconds.

	GDELT	YAGO	WIKI	ICEWS14	ICEWS18	smallp.	polecat	icews	wikidata
total time [s]	21,926	191	143	825	21,149	280	13,681	276,269	43,029
creation of examples [s]	12,244	9	10	111	242	7	1,464	129,139	280
parameter learning [s]	5,670	11	0.4	366	7,743	1	594	2,221	18
rule application [s]	3,704	125	106	279	12,593	194	10,957	137,574	39,911
evaluation [s]	253	45	18	64	510	69	620	6335	2,637
# rules	134,574	8,140	350	53,553	1,010,307	1,221	1,024	167,670	12,050

Table 7: Inference time as well as total train and validation times as reported in (Gastinger et al. 2024) in seconds.

Method	smallpedia		polecat		icews		wikidata	
	Test	Total	Test	Total	Test	Total	Test	Total
EdgeBank$_{tw}$	2,935	5,810	46,629	94,475	311,278	600,929	5,445	8,875
RecurrencyBaseline	310	9,895	3,392	80,378	3,928	148,710	-	-
RE-GCN	165	3,895	1,766	45,877	6,848	114,370	-	-
CEN	331	14,493	2,726	77,953	8,999	202,477	-	-
TLogic	331	803	75,654	138,636	60,413	128,391	-	-

Shilliday, A.; Lautenschlager, J.; et al. 2012. Data for a worldwide ICEWS and ongoing research. *Advances in Design for Cross-Cultural Activities*, 455.

Sun, Z.; Vashishth, S.; Sanyal, S.; Talukdar, P. P.; and Yang, Y. 2020. A Re-evaluation of Knowledge Graph Completion Methods. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 5516–5522.

Vrandečić, D.; and Krötzsch, M. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10): 78–85.